Bubble check: a simplified algorithm for elementary check node processing in extended min-sum non-binary LDPC decoders
Emmanuel Boutillon, Laura Conde-Canencia

To cite this version:

HAL Id: hal-00482730
https://hal.archives-ouvertes.fr/hal-00482730
Submitted on 11 May 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Bubble check: a simplified algorithm for elementary check node processing in extended min-sum non-binary LDPC decoders

E. Boutillon and L. Conde-Canencia

A simplified algorithm for the check node processing of extended min-sum non-binary LDPC decoders is proposed. This novel technique, named bubble check, can reduce the number of compare operations by a factor of three at the elementary check node level. As this significant complexity reduction is achieved without any performance loss, this technique becomes highly attractive for hardware implementation.

Introduction: Non-binary low density parity-check (NB-LDPC) codes are constructed as a set of parity equations over a Galois field GF(q). Even if they are now known to be an efficient alternative to binary LDPC for the transmission of short frames, their major drawback remains their high decoding complexity, especially at the check node processors. In [1], we highlighted the interest of the extended min-sum (EMS) algorithm applied to NB-LDPC [2, 3] because of the significant complexity reduction it introduces compared to the belief propagation (BP) algorithm. To be specific, the \((q \times log\ q)\)-BP-complexity is reduced to \(n_{u} \times log\ n_{u}\), where \(n_{u}\) is the size of the truncated probability messages in the decoder \(n_{u} < q\). However, from a hardware point of view, the EMS complexity is still in the order of \(n_{u}^{2}\). In this Letter, we introduce the bubble check algorithm, a technique that reduces this complexity to the order of \(n_{u}n_{t}/n_{u}^{2}\).

EMS elementary check node (ECN) processing: Let us consider a forward/backward implementation of the node update [4]. The check node equation can then be expressed by several elementary steps, defined by a node update that assumes two input messages \(U, V\) and one output message \(E\). These three messages are log-likelihood-ratios (LLR) sorted in increasing order, i.e. \(U = [U(1), U(2), \ldots, U(n_{u})]\), with \(U(1) = 0\) and \(U(i) = -log[P(U^{(i)})/P(U^{(i-1)})].\) In this equation, \(U_{i}^{(i)}\) is the GF(q) index associated to \(U_{i}\) and \(L_{i}\) are the local hypotheses. Note that the same notation applies for \(V\) and \(E\) and that we consider the opposite of the classical LLR definition, which simplifies the global architecture of the decoder [5].

The EMS generates \(E\), the output vector containing the \(n_{u}\) smallest values in the set \([U(1) + F(j), (i, j) \in [1, n_{u}]^{2}\]. This set can be represented as a matrix \(T_{5}\), where \(T_{5}(i, j) = U(i) + F(j)\). As \(U\) and \(V\) are sorted in increasing order, the elements of \(T_{5}\) have the following property:

\[
\text{Property 1: } \forall (i, j) \in [1, n_{u}]^{2}, \forall (i', j') \in [1, n_{u}]^{2}, i \leq i' \text{ and } j \leq j' \Rightarrow T_{5}(i, j) \leq T_{5}(i', j')
\]

EMS algorithm: To obtain the output vector \(E\), the authors in [3] use a sorter that contains \(n_{u}\) competing elements from \(T_{5}\). This sorter is in fact a subset \(B\) of the elements of the matrix \(T_{5}\) which is initialised with the first column of \(T_{5}\) and dynamically updated at each step of the algorithm, as follows:

1. **For k = 1 to \(n_{u}\) loop**
   
   **Step 1**: Extract the smallest value in \(B\), i.e., \(T_{5}(i, j)\). This element becomes a new element of \(E\).
   
   **Step 2**: Replace the extracted value in the sorter by \(T_{5}(i, j + 1)\).

   To extract the smallest value in one clock cycle, the authors in [3] propose the parallel insertion of the new incoming value in \(B\). Since this operation is performed \(n_{u}\) times, the global complexity of the EMS is dominated by \(n_{u}^{2}\). Note that we omit the calculation of the GF(q) computations that are also performed at the EMS ECN because no novelty is introduced concerning these.

New approach to EMS ECN processing: The principle is to exploit the properties of the values in \(T_{5}\) to minimise the size of the sorter and thus reduce the order of complexity of the EMS. From property 1, it follows that \(T_{5}(1, 1)\) is the minimum value of \(T_{5}\) and it will thus be extracted to occupy the first position of \(E\) (i.e., \(E(1) = T_{5}(1, 1) = 0\)). For the second position of \(E\), there are two candidates: \(T_{5}(2, 1)\) and \(T_{5}(1, 2)\). If, for example, \(T_{5}(2, 1)\) is extracted (i.e. \(E(2) = T_{5}(2, 1) < T_{5}(1, 2)\)), then the two candidates for the third position are \(T_{5}(3, 1)\) and \(T_{5}(1, 2)\), and so on. Note that, for each position, all the candidates belong to a different row and a different column in \(T_{5}\).
and characterised by a fixed variable node degree $d_v = 2$ [6]. We considered horizontal shuffle scheduling, forward/backward processing and $n_m = 16$. Fig. 3 shows simulation results for codewords of length $N = 192$ symbols and rate $R = 1/2$, with $n_b = 2, 3, 4, 5$ and $n_b = \psi(n_m) = 6$. The bubble check presents no performance loss for $n_b \geq 4$. For $n_b = 3$ and $2$, the performance loss is around 0.04 and 0.4 dB, respectively. The simulation of other code lengths and rates (not shown in this Letter) confirms that the performance of the bubble check algorithm with $n_b = 4$ bubbles remains identical to the performance of the EMS algorithm. This shows that, in practice, the complexity can be chosen to be even lower than the theoretical by using $n_b < \psi(n_m)$, without performance loss.

**Fig. 3** Simulation results for $N = 192$, $R = 1/2$, $n_m = 16$ and 20 decoding iterations

‘EMS’ corresponds to EMS ECN algorithm defined in [3]. ‘Bubble check’ is the EMS ECN bubble check presented throughout.

**Conclusion:** The bubble check is presented as an original algorithm for EMS ECN processing of NB-LDPC decoders. We believe that the $\sqrt{n_m}$-complexity reduction it introduces is a key feature for practical implementation.

**Acknowledgments:** This work is supported by INFSCo-ICT-216203 DAVINCI ‘Design And Versatile Implementation of Non-binary wireless Communications based on Innovative LDPC Codes’ (www.ict-davinci-codes.eu), funded by the European Commission under the Seventh Framework Program (FP7).

© The Institution of Engineering and Technology 2010
10 March 2010
doi: 10.1049/el.2010.0566

E. Boutillon and L. Conde-Canencia (Université Européenne de Bretagne, UBS, Lab-STICC, CNRS UMR 3192, Centre de Recherche - BP 92116, Lorient Cedex F-56321, France)
E-mail: laura.conde-canencia@univ-ubs.fr

**References**